UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/015,256	12/13/2001	Mary I. Hageman	YOR9-2001-0721-US1	1723

28211 7590 11/18/2005

FREDERICK W. GIBB, III
GIBB INTELLECTUAL PROPERTY LAW FIRM, LLC
2568-A RIVA ROAD
SUITE 304
ANNAPOLIS, MD 21401

EXAMINER	
CHOJNACKI, MELLISSA M	
ART UNIT	PAPER NUMBER
2164	

DATE MAILED: 11/18/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/015,256
Filing Date: December 13, 2001
Appellant(s): HAGEMAN ET AL.

Mohammad S. Rahman
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed September 1, 2005 appealing from the Office action mailed January 12, 2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

The following is a listing of the evidence (e.g., patents, publications, Official Notice, and admitted prior art) relied upon in the rejection of claims under appeal.

- Sziklai et al. (U.S. Patent No. 6,341,287)
- Java™ Web Start 1.4.2 Release Notes (www.Javasoft.com)

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-21 are rejected under 35 U.S.C. 102(e) as being anticipated by Sziklai et al. (U.S. Patent No. 6,341,287).

As to claim 1, Sziklai et al. teaches a method for tracking custom computer application development profiles in a data processing system (See abstract, where "custom computer application" is read on "information on operations and requirements concerning an activity or area of business"; also see column 8, lines 60-65), the method comprising:

creating the profiles with a first database tool (See column 8, lines 25-41, lines 60-67; column 9, lines 1-3, where "profiles" is read on "business operations"; also see column 9, lines 13-19);

gathering requirements of the profiles with a second database tool (See abstract; column 10, lines 24-33, where "disposal of hazardous waste in landfills" is used as an example of collecting profile data and regulations);

tracking modifications of the profiles with a third database tool (See column 8, lines 65-67; column 9, lines 1-3; column 32, lines 24-34);

allowing security and authorization users access to the profiles (See column 9, lines 13-19; column 14, lines 50-58; column 21, lines 65-67); and

determining whether breaches in security of the data processing system has occurred in each phase of development of a computer application program (See abstract; column 9, lines 10-16; column 14, lines 50-62; column 21, lines 65-67; column 33, lines 5-10; column 34, lines 1-4).

As to claims 2, 9 and 16 Sziklai et al. teaches wherein in the step of tracking modifications of the profiles with a third database tool, the third database tool comprises

a Profile Matrix, wherein the Profile Matrix comprises a data set (See column 13, lines 14-22; column 25, lines 56-67; column 26, lines 1-7); wherein the third database tool comprises a Profile Matrix, and wherein the Profile Matrix comprises a data set (See column 13, lines 14-22; column 25, lines 56-67; column 26, lines 1-7); wherein the third database tool comprises a Profile Matrix, and wherein the Profile Matrix comprises a data set (See column 13, lines 14-22; column 25, lines 56-67; column 26, lines 1-7).

As to claims 3, 10 and 17 Sziklai et al. teaches wherein in the step of tracking modifications of the profiles with a third database tool, the third database tool allows tracking capability of tasks required to gather and implement changes to the profiles (See abstract; column 7, lines 42-57; column 8, lines 25-41; column 9, lines 58-61); wherein the third database tool allows tracking capability of tasks required to gather and implement changes to the profiles (See abstract; column 7, lines 42-57; column 8, lines 25-41; column 9, lines 58-61); wherein in the method, the step of tracking modifications of the profiles with a third database tool allows tracking capability of tasks required to gather and implement changes to the profiles (See abstract; column 7, lines 42-57; column 8, lines 25-41; column 9, lines 58-61).

As to claims 4, 11 and 18 Sziklai et al. teaches wherein in the step of gathering requirements of the profiles with a second database tool, the second database tool comprises a profile requirement worksheet, wherein the profile requirement worksheet identifies the data (See column 9, lines 32-40, where "worksheet" is read on "worklist";

column 10, lines 47-53); wherein the second database tool comprises a profile requirement worksheet, and wherein the profile requirement worksheet identifies the data (See column 9, lines 32-40, where "worksheet" is read on "worklist"; column 10, lines 47-53); wherein the second database tool comprises a profile requirement worksheet, and wherein the profile requirement worksheet identifies the data (See column 9, lines 32-40, where "worksheet" is read on "worklist"; column 10, lines 47-53).

As to claims 5, 12 and 19 Sziklai et al. teaches wherein in the step of gathering requirements of the profiles with a second database tool, the second database tool further identifies authorization objects and field values of the profile requirement worksheet necessary to gather the requirements of the profiles (See column 11, lines 13-22, lines 28-30; column 21, lines 11-15); wherein the second database tool further identifies authorization objects and field values of the profile requirement worksheet necessary to gather the requirements of the profiles (See column 11, lines 13-22, lines 28-30; column 21, lines 11-15); wherein the second database tool further identifies authorization objects and field values of the profile requirement worksheet necessary to gather the requirements of the profiles (See column 11, lines 13-22, lines 28-30; column 21, lines 11-15).

As to claims 6, 13 and 20 Sziklai et al. teaches wherein the step of creating the profiles with a first database tool further comprises editing the profiles (See column 19, lines 30-32; column 21, lines 21-23); wherein the first database tool edits the profiles

(See column 19, lines 30-32; column 21, lines 21-23); wherein in the method, the step of creating the profiles with a first database tool further comprises editing the profiles (See column 19, lines 30-32; column 21, lines 21-23).

As to claims 7, 14 and 21 Sziklai et al., teaches wherein in the step of creating the profiles with a first database tool, the first database tool comprises a security and authorization profile change request database, wherein the security and authorization profile change request database allows the authorization users and requestors an ability to view documented progress of queries of the profiles (See column 11, lines 36-42, where "authorization users and requestors" is read on "configuration user ";column 29, lines 63-64; column 30, lines 17-25); wherein the first database tool comprises a security and authorization profile change request database, and wherein the security and authorization profile change request database allows the authorization users and requestors an ability to view documented progress of queries of the profiles (See column 11, lines 36-42, where "authorization users and requestors" is read on "configuration user ";column 29, lines 63-64; column 30, lines 17-25); wherein the first database tool comprises a security and authorization profile change request database, and wherein the security and authorization profile change request database allows the authorization users and requestors an ability to view documented progress of queries of the profiles (See column 11, lines 36-42, where "authorization users and requestors" is read on "configuration user ";column 29, lines 63-64; column 30, lines 17-25).

As to claim 8, Sziklai et al., teaches a computer system executing a method for tracking custom computer application development profiles in a data processing system (See abstract, where "custom computer application" is read on "information on operations and requirements concerning an activity or area of business"; also see column 8, lines 60-65), the system comprising:

a first database tool (See abstract; column 32, lines 12-21; column 34, lines 5-8);

a second database tool connected to the first database tool (See abstract; column 32, lines 24-31; column 34, lines 5-8);

a third database tool connected to the first and second database tool (See abstract; column 32, lines 32-41; column 34, lines 5-8);

a data bank connected to the first, second and third database tool (See abstract; column 34, lines 5-8); and

a security and authorization interface connected to the data processing system (See column 9, lines 13-19; column 14, lines 50-58; column 21, lines 65-67), wherein the first database tool comprises a first set of protocols which create the profiles (See column 8, lines 25-41, lines 60-67; column 9, lines 1-3, where "profiles" is read on "business operations"; also see column 9, lines 13-19),

wherein the second database tool comprises a second set of protocols which

gather requirements of the profiles (See abstract; column 10, lines 24-33, where "disposal of hazardous waste in landfills" is used as an example of collecting profile data and regulations);

wherein the third database tool comprises a third set of protocols which track modifications of the profiles (See column 8, lines 65-67; column 9, lines 1-3; column 32, lines 24-34); and

wherein the third database tool is adapted to determine whether breaches in security of the data processing system has occurred in each phase of development of a computer application program (See abstract; column 9, lines 10-16; column 14, lines 50-62; column 21, lines 65-67; column 33, lines 5-10; column 34, lines 1-4).

As to claim 15, Sziklai et al., teaches a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform a method for tracking custom computer application development profiles in a data processing system (See abstract, where "custom computer application" is read on "information on operations and requirements concerning an activity or area of business"; also see column 8, lines 60-65), the method comprising:

creating the profiles with a first database tool (See column 8, lines 25-41, lines 60-67; column 9, lines 1-3, where "profiles" is read on "business operations"; also see column 9, lines 13-19);

gathering requirements of the profiles with a second database tool (See abstract; column 10, lines 24-33, where "disposal of hazardous waste in landfills" is used as an example of collecting profile data and regulations);

tracking modifications of the profiles with a third database tool (See column 8, lines 65-67; column 9, lines 1-3; column 32, lines 24-34);

allowing security and authorization users access to the profiles (See column 9, lines 13-19; column 14, lines 50-58; column 21, lines 65-67); and

determining whether breaches in security of the data processing system has occurred in each phase of development of a computer application program (See abstract; column 9, lines 10-16; column 14, lines 50-62; column 21, lines 65-67; column 33, lines 5-10; column 34, lines 1-4).

(10) Response to Argument

In response to applicants' arguments regarding *"Independent claims 1 and 15 contain features, which are patentably distinguishable from the prior art references of record, and in particular Sziklai. Specifically, claims 1 and 15 provide, in part, '...determining whether breaches in security of said data processing system has occurred in each phase of development of a computer application program'",* the arguments have been fully considered but are not found to be persuasive, because Sziklai et al. discloses a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Claims 1 and 15 allow as

little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding independent claims 1 and 15, that *"Sziklai is and should properly be considered a non-enabling reference, and thus should not be considered as prior art for the purpose of teaching 'determining whether breaches in security of said data processing system has occurred in each phase of development of a computer application program'...Here, as admitted in the Office action (see page 9), Sziklai does not 'does not go into full detail of 'determining security breaches". Thus, Sziklai does not properly enable one skilled in the art to make and use the invention in Sziklai for the purposes of the manner and approach of determining whether security breaches occur. Furthermore, with respect to determining whether security breaches occur, clearly Sziklai does not place the claimed invention within the possession of the public as required by well-established and legally binding case law."* The arguments have been fully considered but are not found to be persuasive, because all Patented Patents are considered enabling. Also, Sziklai et al. does disclose detecting security breaches by disclosing a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes"

[http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Claims 1 and 15 allow as little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Furthermore, Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding independent claims 1 and 15, that *"Additionally, even If by some inexplicable chance that one of ordinary skill in the art would refer to Sziklai and conclude that it sufficiently teaches determining whether security breaches occur, it would be a gross stretch of reason that one of ordinary skill in the art would conclude that Sziklai sufficiently teaches determining whether breaches in security of a data processing system has occurred in each phase of development of a computer application program. There is simply no teaching of this aspect or reasonable interpretation of the broad non-enabling concepts described in Sziklai of this second aspect of the claimed feature".* The arguments have been fully considered but are not found to be persuasive, because claims 1 and 15 allow as little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Regardless, the evidence shows that the "Java security model", as disclosed by Sziklai et al. does teach automatically detecting "registered" and "unregistered" JRE's for each

JAWS application (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). Sziklai et al.'s "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application.

In response to applicants' arguments regarding independent claims 1 and 15, that *"Again, the features relating to security are novel features not taught in Sziklai. Specifically, column 9, lines 13-16 of Sziklai only generically refers to the fact that security is an important feature in database management. There is no mention of how such an implementation of security is to take place, let alone a determination of when breaches in security occur in the development of a software program (i.e., computer application program). That is, there is nothing in this language that suggests that determining whether breaches in security of a data processing system has occurred in each phase of development of a computer application program".* The arguments have been fully considered but are not found to be persuasive, because Sziklai et al. discloses a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically

checking for security breaches for each JAWS application. Claims 1 and 15 allow as little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding independent claims 1 and 15, that Sziklai *"does not suggest, and no logical interpretation of this would suggest that the Java framework implemented by Sziklai can determine whether security breaches have occurred in all phases of the development of a software program. Furthermore, column 21, lines 65-67 of Sziklai once again very generically, and in non-enabling language, establishes implementing a security role to grant/restrict access to the database".* The arguments have been fully considered but are not found to be persuasive, because Sziklai et al. discloses a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Claims 1 and 15 allow as little as one "phase of development", Sziklai et al. teaches a program with at least one

phase of development. Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding independent claim 8, that *"Independent claim 8 contains features, which are patentably distinguishable from the prior art references of record, and in particular Sziklai. Specifically, claim 8 provides, in part, '...determining whether breaches in security of said data processing system has occurred in each phase of development of a computer application program'. These features are simply not taught or suggested in Sziklai",* the arguments have been fully considered but are not found to be persuasive, because Sziklai et al. discloses a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Claim 8 allows as little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other

relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding independent claim 8, that *"Sziklai is and should properly be considered a non-enabling reference, and thus should not be considered as prior art for the purpose of teaching 'to determine whether breaches in security of said data processing system has occurred in each phase of development of a computer application program'" ...Here, as admitted in the Office action (see page 9), Sziklai does not 'does not go into full detail of 'determining security breaches". Thus, Sziklai does not properly enable one skilled in the art to make and use the invention in Sziklai for the purposes of the manner and approach of determining whether security breaches occur. Furthermore, with respect to determining whether security breaches occur, clearly Sziklai does not place the claimed invention within the possession of the public as required by well-established and legally binding case law."* The arguments have been fully considered but are not found to be persuasive, because all Patented Patents are considered enabling. Also, Sziklai et al. does disclose detecting security breaches by disclosing a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application.

Furthermore, Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding independent claim 8, that "*Additionally, even If by some inexplicable chance that one of ordinary skill in the art would refer to Sziklai and conclude that it sufficiently teaches determining whether security breaches occur, it would be a gross stretch of reason that one of ordinary skill in the art would conclude that Sziklai sufficiently teaches determining whether breaches in security of a data processing system has occurred in each phase of development of a computer application program. There is simply no teaching of this aspect or reasonable interpretation of the broad non-enabling concepts described in Sziklai of this second aspect of the claimed feature".* The arguments have been fully considered but are not found to be persuasive, because claim 8 allows as little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Regardless, the evidence shows that the "Java security model", as disclosed by Sziklai et al. does teach automatically detecting "registered" and "unregistered" JRE's for each JAWS application (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). Sziklai et al.'s "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application.

In response to applicants' arguments regarding independent claim 8, that *"Again, the features relating to security are novel features not taught in Sziklai. Specifically, column 9, lines 13-16 of Sziklai only generically refers to the fact that security is an important feature in database management. There is no mention of how such an implementation of security is to take place, let alone a determination of when breaches in security occur in the development of a software program (i.e., computer application program). That is, there is nothing in this language that suggests that determining whether breaches in security of a data processing system has occurred in each phase of development of a computer application program".* The arguments have been fully considered but are not found to be persuasive, because Sziklai et al. discloses a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Claim 8 allows as little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding independent claim 8, that Sziklai *"does not suggest, and no logical interpretation of this would suggest that the Java framework implemented by Sziklai can determine whether security breaches have occurred in all phases of the development of a software program. Furthermore, column 21, lines 65-67 of Sziklai once again very generically, and in non-enabling language, establishes implementing a security role to grant/restrict access to the database".* The arguments have been fully considered but are not found to be persuasive, because Sziklai et al. discloses a "Java security model" which prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Claim 8 allows as little as one "phase of development", Sziklai et al. teaches a program with at least one phase of development. Sziklai et al. also discloses "security implements setting up user groups, system privileges, database privileges and other relevant security activities" (See column 21, lines 65-67), therefore restricting user access to the business information/profiles.

In response to applicants' arguments regarding dependent claims 2, 9, and 16, that Sziklai does not teach *"...said third database tool comprises a Profile Matrix, and*

wherein said Profile Matrix comprises a data set", the argument has been fully considered but is not found to be persuasive, because Sziklai et al. discloses a "report matrix table" that provides matrix reports and a "module tem table" that provides business data from entry form, report and document defined in the system (See column 13, lines 14-22). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In response to applicants' arguments regarding dependent claims 3, 10, and 17, that Sziklai does not teach *"...said third database tool allows tracking capability of tasks required to gather and implement changes to said profiles",* the argument has been fully considered but is not found to be persuasive, because Sziklai et al. discloses "tracking and managing regulatory compliance, non-regulatory requirements and other change intensive business activities" (See column 9, lines 58-67). Sziklai et al. discloses tracking changes to business information/profiles by "absorbing database and application changes that arise from changes in regulations, policies, procedures, processes, materials, and similar factors" (See column 9, lines 58-67; column 10, lines 1-8). Every change is a new modification to the business information database.

In response to applicants' arguments regarding dependent claims 4, 11, and 18, that Sziklai does not teach "...said second database tool comprises a profile requirement worksheet, and wherein said profile requirement worksheet identifies said data", the argument has been fully considered but is not found to be persuasive, because Sziklai et al. discloses "worklists" that can include "modules (data entry forms,

reports and documents), process and sub-worklists" (See column 11, lines 9-13). Therefore, Sziklai et al.'s "worklists" does identify data because modules, process and sub-worklists contain data.

In response to applicants' arguments regarding dependent claims 5, 12, and 19, that Sziklai does not teach *"...said second database tool further identifies authorization objects and field values of said profile requirement worksheet necessary to gather said requirements of said profiles",* the argument has been fully considered but is not found to be persuasive, because Sziklai et al. discloses being able to change fields (See column 11, lines 14-22, lines 28-30) and he also discloses a "worklist table" that provides definitions, logic for worklists and calculations. He further discloses "a calculation profile value table" that records profile variable value calculations (See column 13, lines 24-32). Therefore, the second database, which is the collection of business information, does identify field values in the worklist. Furthermore, the "Java security model" of Sziklai et al. prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Therefore the "Java security model" identifies the "authorization objects".

In response to applicants' arguments regarding dependent claims 6, 13, and 20, that Sziklai does not teach *"...said step of creating said profiles with a first database tool further comprises editing said profiles"*, the argument has been fully considered but is not found to be persuasive, because Sziklai et al. discloses editing reports and "form fields" within the business information/profiles (See column 19, lines 30-32; column 21, lines 21-23). Sziklai et al.'s invention discloses managing data that is constantly changing and detecting those changes, therefore it suffice to say that the business information/profiles is constantly being edited automatically and manually by the user as disclosed by Sziklai et al..

In response to applicants' arguments regarding dependent claims 7, 14, and 21, that Sziklai does not teach *"...said first database tool comprises a security and authorization profile change request database, wherein said security and authorization profile change request database allows said authorization users and requestors an ability to view document progress of queries of said profiles"*, the arguments have been fully considered but are not found to be persuasive, because Sziklai et al. discloses a "advanced query builder", "query editor" and a "query filter". A user uses the "advanced query builder" to create a view (See 11, lines 34-40), the "query editor" receives the users request and takes the user through all the steps required to build a query and the "query filter" apply's restrictions on data (See column 30, lines 17-32). The "Java security model" of Sziklai et al. prevents unauthorized tampering. Further evidence of the "Java security model", as disclosed by Sziklai et al., shows that "Every time JAWS is run, it automatically detects all 'registered' JREs on the computer" (See "Java™ Web

Start 1.4.2 Release Notes" [http://java.sun.com/j2se/1.4.2/docs/guide/jws/relnotes.html], "Automatic detection of JREs" section). This evidence shows that the "Java security model" automatically detects "registered" and "unregistered" JREs on the computer, therefore automatically checking for security breaches for each JAWS application. Therefore the users that are using the "advance query builder" and the "query editor" have passes the "Java security model".

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Mellissa M. Chojnacki

Conferees:

Safet Metjahic

SAFET METJAHIC
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Charles Rones

SAM RIMELL

CHARLES RONES
SUPERVISORY PATENT EXAMINER

SAM RIMELL
PRIMARY EXAMINER

APPEAL CONFERENCE HELD
NOVEMBER 8, 2005. AGREEMENT
TO PROCEED TO BOARD OF APPEALS